| **Name : Parth Kadam** | **Class/Roll No. :D11AD/27** | **Grade :** |
| --- | --- | --- |

**Title of Experiment :** To understand Version Control System / Source Code Management, install git and create a GitHub account

**Objective of Experiment :** To be aware of different Version Control tools like GIT, CVS or Mercurial

**Outcome of Experiment :** To obtain complete knowledge of the ―version control system to effectively track changes augmented with Git and GitHub

**Theory :**

### **Version Control System (VCS):**

A Version Control System is a tool that helps manage changes to source code over time. It enables multiple developers to collaborate on a project by tracking and controlling different versions of files. The primary goals of a VCS include:

* History Tracking: VCS records every change made to the source code, allowing developers to review the history of a project. This includes who made each change, when it was made, and what the changes were.
* Collaboration: VCS facilitates collaboration among team members. Multiple developers can work on the same project simultaneously without interfering with each other's work. The system merges changes intelligently, resolving conflicts when they occur.
* Branching and Merging: VCS allows developers to create branches, which are independent lines of development. This is useful for working on features or bug fixes without affecting the main codebase. Merging combines changes from different branches.
* Revert to Previous States: Developers can revert to a previous version of the code if a mistake is made or if a particular version is needed.
* Backup and Recovery: VCS serves as a backup mechanism. Even if the local copy of code is lost or corrupted, the entire history can be reconstructed from the repository.

### **Source Code Management (SCM):**

Source Code Management is a broader term that encompasses the processes and tools used to manage the source code of a software project. VCS is a critical component of SCM, and SCM also includes practices related to code review, continuous integration, and release management. Key aspects of SCM include:

* Code Review: The practice of having one or more developers review the code changes made by others. This ensures code quality, adherence to coding standards, and knowledge sharing within the team.
* Continuous Integration (CI): CI involves automatically building and testing code changes as soon as they are committed to the version control system. This helps identify issues early in the development process.
* Release Management: Managing the process of creating and delivering software releases. SCM ensures that releases are well-documented, tested, and can be reproduced.
* Configuration Management: SCM helps manage the configuration of a software system. It involves controlling changes to system components and ensuring consistency across different environments.

### Installing Git:

1. Download the Git installer from the official Git website: [Git for Windows](https://gitforwindows.org/).
2. Run the downloaded installer.
3. Follow the installation wizard instructions. The default options are usually sufficient.

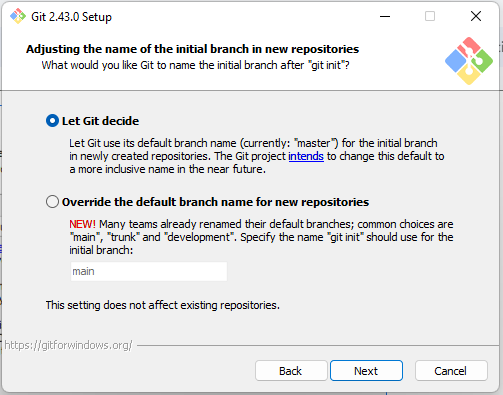
### Creating a GitHub Account:

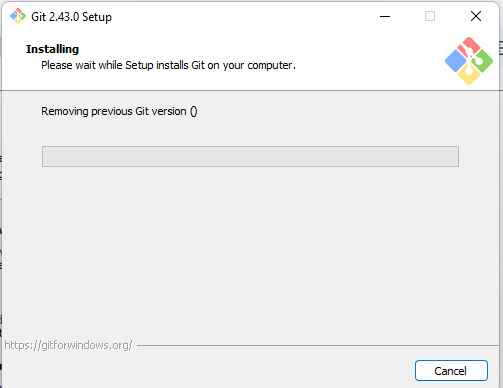
1. Open your web browser and go to [GitHub](https://github.com/).
2. Click on the "Sign up" button in the upper-right corner.
3. Fill out the required information on the sign-up page, including your username, email address, and password. Choose a strong password for security.
4. Complete the "Verify your account" step by solving the puzzle or entering the code sent to your email address.
5. Choose a plan: GitHub offers both free and paid plans. For most users, the free plan is sufficient. Click on "Continue" for the free plan.
6. Optionally, complete the survey or skip it.
7. Choose your preferences for receiving updates from GitHub and click "Submit."
8. Once your account is created, you'll be prompted to set up a new repository. You can either create a repository immediately or skip this step and do it later.

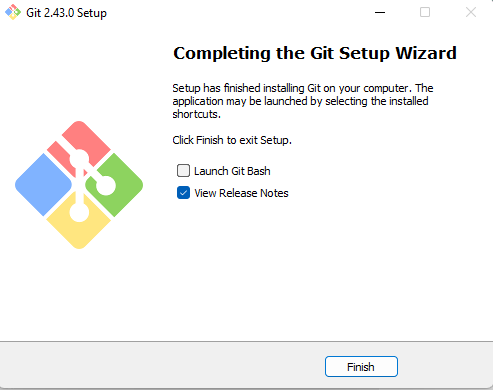
**OUTPUT :**

****

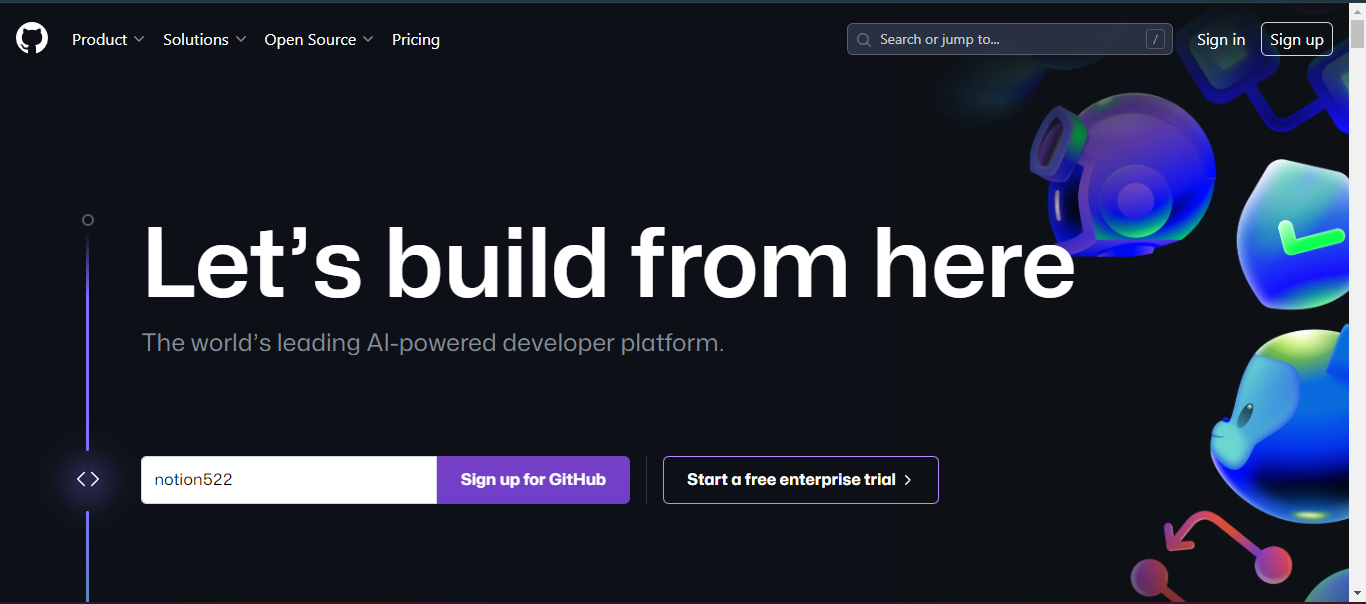


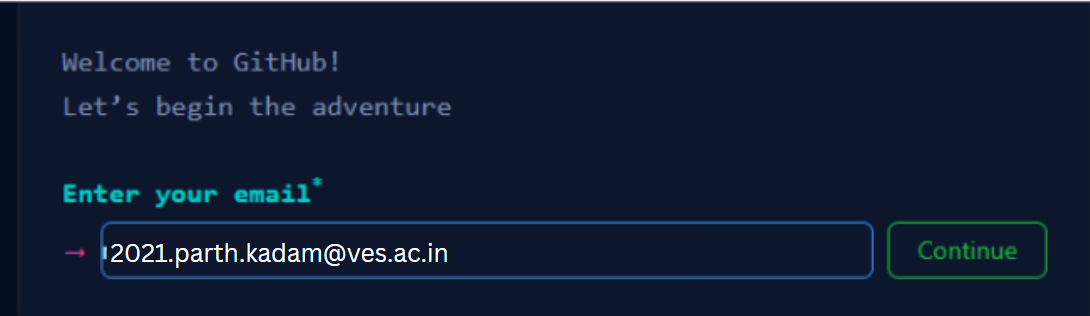


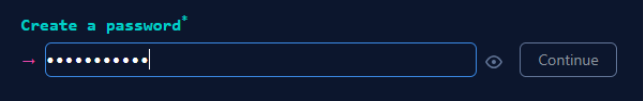




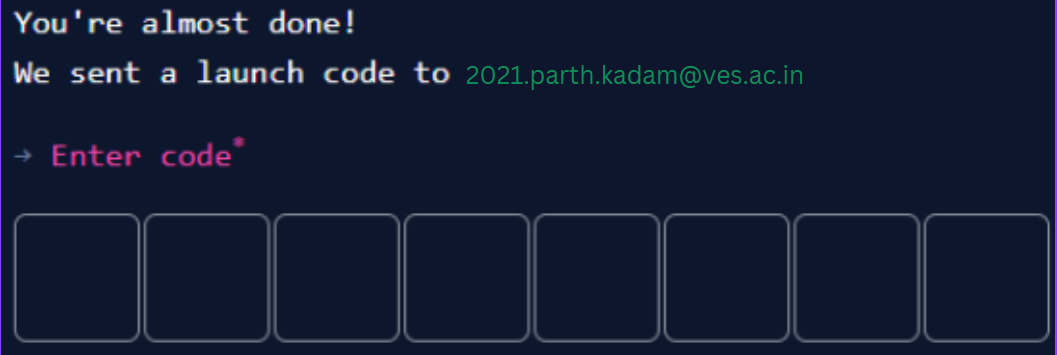
**Creating a github account :**

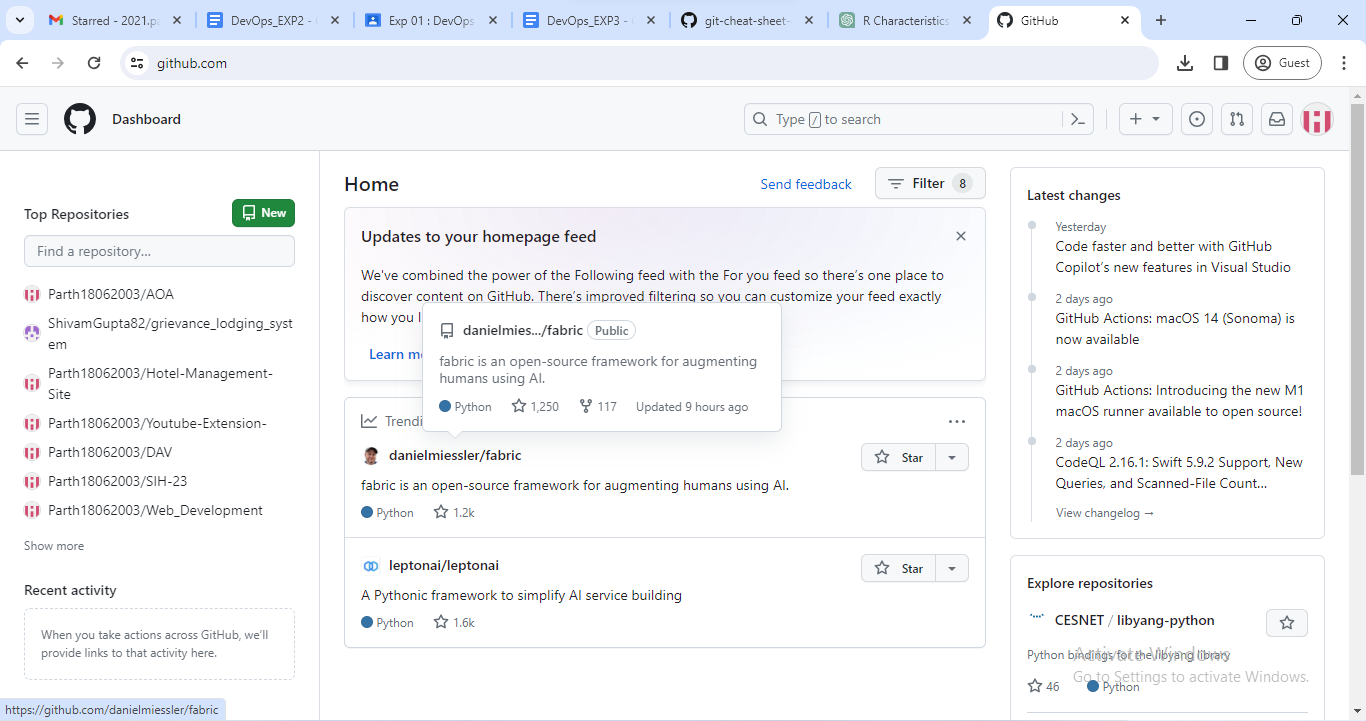










****

**Conclusion :**

Thus we got to know more about VCS and SCM, where the below are somethings that were important to understand in this experiment

* Tracking Changes: VCS keeps a record of every change made to the code. It logs who made the change, when it happened, and what exactly was modified. This helps team members stay organized and aware of what's happening in the project.
* Collaboration: VCS allows multiple developers to work on the same project simultaneously without causing conflicts. It helps integrate everyone's changes seamlessly.
* Code Quality: SCM ensures that the code remains clean, well-organized, and follows coding standards. It acts like a quality check, making sure the code is reliable and easy to understand.
* Reliability: By using VCS and SCM, developers can be confident that the software they're building is dependable. It reduces the chances of errors, making the final product more reliable.